The University of Georgia
Department of Computer Science

Course Information Sheet
CSCI 2610
Discrete Mathematics for Computer Science

Brief Course Description (50-words or less)

This course presents a survey of the fundamental mathematical tools used in computer engineering: sets, relations, and functions; propositional and predicate logic; proof writing strategies such as direct, contradiction and induction; summations and recurrences; elementary asymptotics and timing analysis; counting and discrete probability; undirected and directed graphs with applications in computer science.

Extended Course Description / Comments

CSCI majors should enroll in this course MATH/CSCI 2610 instead of CSCI 2611.

Pre-Requisites and/or Co-Requisites

MATH 1113: Pre-Calculus
This course cannot be taken for credit if student has received credit for MATH/CSCI 2611

Approved Textbooks

Kenneth H. Rosen
Discrete Mathematics and its Applications
7th Edition
ISBN-13: 978-0073229720

Stein, Drysdale and Bogart
Discrete Mathematics for Computer Scientists
1st Edition
ISBN-13: 978-0-13-212271-9

Learning Outcomes

This course presents a survey of topics in discrete mathematics most relevant to students studying computer engineering. At the end of the semester, all students will be able to do the following:

1. Build truth tables for propositional expressions.
2. Prove properties using a variety of proof strategies including direct proofs, proofs by contradiction, proofs by cases, and inductive proofs.
3. Convert a number from one base to another (e.g. from decimal to binary)
4. Add two binary numbers.
5. Recursively define Boolean algebra.
6. Create a Boolean function from a truth table.
7. Create a circuit diagram that computes a given Boolean function
8. Use K-maps to minimize the number of states in a circuit.
9. Use permutations and combinations to count the number of elements in large sets.
10. Apply the pigeonhole principle.
11. Determine conditional probabilities
12. Determine if a function is an injection, a surjection, a bijection, or none of these.
13. Use bijections to prove if a given set is countable
14. Use diagonalization to prove a given set is uncountable.
15. Given an equivalence relation R over a domain D, partition D into subsets (equivalence classes) according to R.
16. Find the degree of a vertex in a graph.

17. Determine if a given graph is a tree.

Relationship Between Student Outcomes and Learning Outcomes

Learning Outcomes	*Student Outcomes*										
	a	b	c	d	e	f	g	h	i	j	k
1	•									•	
2	•									•	
3	•									•	
4	•									•	
5	•										
6	•									•	
7	•									•	
8	•									•	
9	•									•	
10	•									•	
11	•									•	
12	•										
13	•									•	
14	•									•	
15	•									•	
16	•										
17	•										

Student Outcomes

a. An ability to apply knowledge of computing and mathematics appropriate to the discipline.
b. An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution.
c. An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs.
d. An ability to function effectively on teams to accomplish a common goal.
e. An understanding of professional, ethical, legal, security and social issues and responsibilities.
f. An ability to communicate effectively with a range of audiences.
g. An ability to analyze the local and global impact of computing on individuals, organizations, and society.
h. Recognition of the need for and an ability to engage in continuing professional development.
i. An ability to use current techniques, skills, and tools necessary for computing practice.
j. An ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices.
k. An ability to apply design and development principles in the construction of software systems of varying complexity.

Major Topics Covered (Approximate Course Hours)	Propositional logic (3.5-hours) Predicate logic (4-hours) Proofs: types of proofs (4-hours) Sets, set logic and set operations (2-hours) Functions (2-hours) Boolean algebra (3.5-hours) Sequences and summations (2-hours) Integer algorithms (3-hours) Modular arithmetic (1-hour) Mathematical induction (4-hours) Counting (2.5-hours) The pigeonhole principle (.5-hours) Permutations and combinations (3.5-hours) Finite probabilities (4-hours) Relations (2.5-hours) Graph basics (2-hours)
Assessment Plan for this Course	Each time this course is offered, the class is initially informed of the Course Outcomes listed in this document, and they are included in the syllabus. At the end of the semester, an anonymous survey is administered to the class where each student is asked to rate how well the outcome was achieved. The choices provided use a 5-point Likert scale containing the following options: Strongly agree, Agree, Neither agree or disagree, disagree, and strongly disagree. The results of the anonymous survey are tabulated and results returned to the instructor of the course. The course instructor takes the results of the survey, combined with sample student responses to homework and final exam questions corresponding to course outcomes, and reports these results to the ABET committee. If necessary, the instructor also writes a recommendation to the ABET committee for better achieving the course outcomes the next time the course is offered.
How Data is Used to Assess Program Outcomes	Each course Learning Outcome, listed above, directly supports one or more of the Student Outcomes, as is listed in "Relationships between Learning Outcomes and Program Outcomes". For CSCI 2610, Student Outcomes (a) and (i) are supported.
Course Master	Dr. Shelby Funk
Course History	